APPENDIX H

1.0 PREPROCESSING

1.1 DATA PLOTTING

The Brawler Aerodynamics Data Plotting Package (*plmain*) is a program which reads the aircraft (**ACFTD**) and stores (**STORED**) data files and produces plots of various measures of physical characteristics and aerodynamic performance. All graphics subroutines within the package utilize the pre-existing Brawler graphics interface routines. The package produces three types of plots: contour, data table, and IR.

A contour plot consists of a series of contours for user-defined values of the variable of interest. The user is requested to enter a list of constant values for the contour variable. In addition, some plots require the user to enter one or more fixed values to further define the aerodynamic environment for which the plot is to be made. Curve smoothing is performed on all contour plots.

A data table plot produces curves taken directly from the input data in the performance section of the aircraft data file. Curve smoothing is not performed on data table plots.

An IR plot consists of a polar plot of either range versus aspect for a fixed value of probability of detection or signal-to-noise versus aspect for a fixed value of range. The plots require the user to input parameters that specify the desired atmospheric and terrain environment as well as fixed values to further define the aerodynamic environment. Curve smoothing is not performed on the IR plots.

1.2 DATA DESCRIPTION

The aircraft and stores description utility, *dsmain*, is a program that gives users a text translation of the contents of the **ACFTD** and **STORED** input files. The user is prompted for the name of the aircraft or store as well as a keyword identifying the type. To identify an aircraft the keyword is ‘ACFT’ and for stores it is one of the following: ‘EXP’, ‘GUN’, ‘HMS’, ‘IFF’, ‘IRST’, ‘MAW’, ‘MISL’, ‘MW’, ‘NJ’, ‘POD’, ‘PYLN’, ‘RDR’, ‘RWR’, ‘SAN’, ‘SFD’, ‘SJ’, ‘TANK’. ‘ALL’ followed by ‘ACFT’ or one of the above store types causes the program to give a text translation of all aircraft or all stores of the indicated type. This utility produces an output file, **DESCRB**, containing tables of data and data descriptions for the requested aircraft and/or stores.

1.3 MISSILE ENVELOPE PLOTS

The missile envelope program, *evmain*, performs “kinematic” missile envelope calculations and subsequent plot generation. It provides the capability to generate missile envelope plots which are specific to the missile type, envelope algorithm, and engagement geometry under consideration. Two types of plots can be generated by *evmain*:

- Maximum launch range, R_{max}, versus target aspect angle
- Minimum launch range, R_{min}, versus target aspect angle

The envelopes generated by *evmain* are the envelopes used by the pilot in making weapon selection and firing decisions. Plots of R_{max} vs. aspect which are based upon actual missile flyouts may be generated using the envelope generator program *egmain*.

Each execution of the missile envelope generator produces two polar plots that display R_{min} and R_{max} as a function of target aspect angle, respectively. R_{min} is displayed in units of feet, while R_{max} is displayed in units of nmi.

1.4 MISSILE KINEMATIC ENVELOPE GENERATION

The envelope generator program *egmain* provides the user with the capability to generate plots of R_{max} or R_{min} vs. aspect *based upon actual missile flyouts* for any missile in the database. This capability is in contrast to the missile envelope utility *evmain*, described above, which plots the envelope specification, which is a part of the missile data set. The program operates in two modes. In the first mode, a table of range (minimum or maximum) vs. aspect is constructed and written to disk; in the second mode, the table is read in and displayed on a graphics terminal as a range versus aspect polar plot. Although both modes could have been combined, by separating them, the user can construct the table in batch or background mode without tying up a terminal for the graphical output. In the first mode, or generation mode, the program attempts to find R_{max} (or R_{min}), to within a user-specified tolerance for each aspect by repetitively running a Brawler engagement iterating on range until R_{max} (or R_{min}) can be determined to that tolerance. An engagement is considered a success if the missile flies to a point at which it attempts to fuze using the fuzing algorithm appropriate to the missile, either point of closest approach or fuzing cone. Endgame can be constrained by specifying a non-zero value for missile “gee” capability at endgame. The missile is then required to be able to pull at least this number of gees at endgame in order for a successful endgame to be declared. A table is then written to the output file **EGOUT** containing range vs. aspect information. When the program is invoked in the second, or plot mode, this file is read and the data displayed in a polar plot of range vs. aspect.

1.5 E-POLE GENERATION

The E-pole program, *epmain*, provides the user with the capability to determine the range to a hostile aircraft, initially approaching nose-on, at which initiation of an escape maneuver will insure that any missile fired by the aircraft (any time after the maneuver is begun) cannot reach the maneuvering aircraft. Since it is not feasible to fire a continuous stream of missiles, the program fires 3 per engagement:

a. Fire missile “simultaneously” with maneuver initiation.

b. Fire missile as maneuvering aircraft passes through aspect of 80˚ off the tail.

c. Fire at the earlier of 1) time when negative closure occurs between aircraft, or 2) a user-specified time since the second missile firing.

In order to determine the escape range, the program iterates on the range at which the maneuver is initiated until it is within a user-specified tolerance of the minimum escape range. For efficiency, each iteration terminates when either a missile achieves fuzing criteria or the last (3^{rd}) missile is removed from the simulation.

Both pre-launch fire control constraints and post-launch signal requirements are played for the missiles being fired. Thus IR missiles which require lock-up on the rail must see sufficient signal, command-guided missiles requiring radar tracks must have them established, etc. For this reason each iteration is begun at a range 30,000 ft. greater than the desired first missile firing/escape initiation range so that avionics constraints may be met on the launching aircraft.

The program does not constrain the analyst to use the same type of missile for each of the three shots. For example, the analyst can specify a radar missile for the nose and tail shots and an IR missile for the beam shot since it may not be possible to lock-up the radar on the beam. In the input data, the user specifies the missile fox number for each shot.

1.6 F-POLE GENERATION

The F-pole program provides the user with the capability to collect F-pole, A-pole and evasion time information from head-on 1v1 engagements. F-pole is defined as the distance from the launcher to the target when the missile achieves endgame. A-pole is the distance from the launcher to the target when the missile acquires the target (only defined if the first phase of the missile is command guided). Evasion time is the time between first missile endgame and second missile endgame (only defined if both missiles reach endgame).

The F-pole program is set up as a 1v1 engagement with a single missile loaded on each aircraft. The geometry is set up so that the aircraft are flying head on, at the altitude and speed given in the input file **FPINPT**. *fpmain* runs a separate case for each combination of desired launch ranges for the blue and red aircraft specified in the **FPINPT** input file. Each case of an *fpmain* run is very similar to a Brawler run; *fpmain* was built from Brawler and only minor changes were introduced to allow the desired information to be collected.

Both pre-launch fire control constraints and post-launch signal requirements are played for the missiles being fired. Thus, IR missiles which require lock-up on the rail must see sufficient signal, command-guided missiles requiring radar tracks must have them established, etc. Each iteration is begun at a range 30,000 ft. greater than the desired first missile launch range so that avionics constraints may be met on the launching aircraft. Three variations are allowed in the level of detail associated with the signal detection algorithms in the seeker:

a. The normal Brawler signal level algorithms are used.

b. Detections are declared whenever the probability of detection exceeds 50%.

c. The probability of detection in the seeker is set to 1.0.

One missile is fired by each aircraft during each case of the F-Pole program. Several artificial constraints have been added for the F-Pole program to allow the desired information to be collected:

a. All missile P_k's are set to zero. This allows the F-pole to be calculated for the second missile that reaches endgame.

b. The probability of successful launch and the probability of successful guidance initiation for each missile are overridden and are set to 1.0.

c. Endgame is defined as meeting fuzing conditions.

For each aircraft, the user specifies the type of maneuver to be performed for each of four phases: Pre-launch, pre-acquisition, post-acquisition, and post-endgame. For the pre-launch phase, the user must indicate when the maneuver is to begin expressed as a range to the desired launch point. All four phases must be specified even if the missile being used acquires on the rail. In this case the aircraft will enter the pre-acquisition phase momentarily at launch and transition immediately to post-acquisition.

3.0 POSTPROCESSING

3.1 GRAPHICS

The graphical output capability of the Brawler simulation system adds a powerful tool to the analyst's arsenal by enhancing his ability to develop models and by increasing confidence in his results. *grmain* is a stand-alone interactive graphics program which uses as input the history file generated by Brawler.

Some of the features of the program which provide ease of use and interpretation are:

- Simple command structure.
- Color graphics (if terminal and software capable).
- Capability to easily switch between plot types.
- Ability to read commands from disk files.
- Ability to proceed forward and backward in time.
- Visible horizon and latitude-longitude grid.
- Hidden line algorithm.
- Ability to divert graphics structures to disk files (machine dependent).
- Ability to read graphics structures from disk files (machine dependent).

The program can receive input from various sources: the terminal keyboard, history files, and command files. History files, which are generated by the execution of Brawler, contain time-sequential histories of the simulation.

Three types of graphical output are currently available:

- COCKPIT Plot.
- VEWBATL Plot.
- TRAIL Plot.

The COCKPIT plot produces a pilot's eye view of the air battle with a view of other aircraft, missiles, and an earth surface grid. The VEWBATL plot generates an overview of the air battle in the horizontal plane with aircraft and missiles represented symbolically. The TRAIL plot provides a view from an observation point east or south of the battle with the observers elevated at an arbitrary angle from the horizontal.

Alphanumeric Displays

The graphics screen is divided into three spatial fields:

- Title Field

- Data Field.
- Graphics Field.

The Title Field displays information which will uniquely identify the plot by type and associate it with a particular Brawler run.

The Data Field, along the left boundary of the display, contains various blocks of user-selected numeric information.

Within the Data Field, nine types of alphanumeric displays are available. The display segments may be printed with any plot type with the exception of attacker and target data which are restricted to COCKPIT plots. The contents of each segment type is listed in Table 8.1-2.

TABLE 8.1-2. Data Field Printing Options.

Display Segment	For	Displays	Units
Score (*score_print*)	Each side	Number of aircraft alive. Number of aircraft killed.	-
Status (*status_print*)	Each aircraft	Aircraft index. Alive/dead. Inertial position (x, y, z). Mach number. Rate of climb.	- - (nmi, nmi, kft) - kft/min
Missiles (*msl_print*)	Each live or recently expired missile	Label. Unique I.D. Target aircraft. Range to target (if live). Range rate (if live). Terminal status (if dead).	- octal - kft kft/sec -
Situation (*situation_print*)	Each live aircraft	Aircraft index. Posture. Target aircraft. Missile mode.	- - - -
Attacker (*attacker_print*)	Viewing aircraft only	Aircraft index. Mach number. Altitude. g's Angle of attack. Heading from north. Climb angle. Roll angle. Pitch angle. Roll rate.	- - kft - deg deg deg deg deg/sec deg/sec
Targets (*target_print*)	All other aircraft	Aircraft index. Mach number. Relative range. Relative elevation angle. Relative azimuth angle.	- - kft deg deg

TABLE 8.1-2. Data Field Printing Options.

Display Segment	For	Displays	Units
Present State (*prestate_print*)	Live aircraft	Aircraft index. Altitude. Calibrated airspeed. g's. Mach. Rate of climb.	- kft kts - - kft/min
Decision Levels (*dlevelp*)	Each aircraft	Current alternative at selected decision level.	-
Aircraft Alternatives (*acaltp*)	Each aircraft on list	Current alternatives at all levels.	
Phases (*phase_print*)	Each aircraft	Current phase of engagement	
Mental Model (*mental_model_print*)	1 aircraft	List of a/c and missiles in mental model.	

2.2 STATISTICS REPORTS

The Measures of Performance (MOP) package is a subsystem of the Brawler air combat model used for reporting certain statistics about sets of Brawler engagements. The MOP system maintains a database to which data is appended during each Brawler run. At the conclusion of a set of engagements, the database can be accessed by the MOP reporting package to compute engagement set statistics and print the results.

Output is sent to one of two files, depending on the phase of the process which is being conducted. **MPDB**, which holds the MOP database, is created during an initialization phase and updated during successive Brawler engagements to reflect the statistics gathered during each run. During the final report phase, **MPDB** is used as input to the report generator *mrmain*, which generates a printable file, **MOPOUT**. The appearance of the report in **MOPOUT** is controlled by input from **MPCTRL**, which includes a list of statistics on which to report. A detailed description of these statistics is included in the Analyst Manual, Section 3.2.4.

Pages 1 and 2 of the MOP report file **MOPOUT** contain diagnostic output reflecting the data input from the file **MPCTRL**. Under most circumstances this page can be ignored; it is designed so that a page eject occurs following it so that it can be torn off and discarded.

Page 3 is the first report page. This page contains overall titling information, a general description of the group of scenarios on which the report is based (read in free-format from **MPCTRL**), and the values of a few important, scenario-wide parameters which define the scenarios.

Page 4 provides additional definition of the scenario groups, giving a list of the home bases and some data on each aircraft involved. Note that if more than one scenario is being used, this data is for <u>only the first scenario requested</u>. Finally, one line concerning each engagement over which statistics are being aggregated is printed, including the scenario the engagement is from and the random number seed.

Page 5 offers little information that is not offered on subsequent pages, but the data on page 4 is formatted here for cleaner presentation. An attrition summary and weapon summary (by azimuth sector) is also given on this page.

The remaining pages in the document are individual statistical measures, as requested through input from the **MPCTRL** file. As appropriate, statistics are broken out by aircraft category, by weapon or detection category, and perhaps by azimuth sector as well. The confidence level columns currently have a well-defined meaning only for statistics whose distribution is normal (exchange rate is an exception; its confidence levels are well-defined). The equations describing each mean and standard deviation are given in the Analyst Manual Section 3.2.5.